Ground Penetrating Radar
Volume 2, Issue 1, March 2019
https://doi.org/10.26376/GPR2019003

REAL TIME VISUALIZATION OF THE DATA GATHERED BY A RECONFIGURABLE STEPPED FREQUENCY GPR SYSTEM

FILIPPO BRIGATTI

Marketz SpA, Milano, Italy – filippobrigatti@virgilio.it

ABSTRACT

This paper describes recent improvements made to the acquisition software of a reconfigurable stepped-frequency ground penetrating radar (GPR) prototype, to allow real-time data visualization. In particular, real-time data visualization was not yet implemented in the previous version of the acquisition software, although this is a common feature available in all commercial systems. This was a bad problem for the GPR prototype: the possibility to visualize data in real time is obviously of vital importance, because it makes it possible for the user to easily identify promising areas in the field, or to recognize an anomalous functioning of the system without wasting a day of work. So far, real-time data visualization was not yet possible because the prototype at hand is equipped with three equivalent couples of antennas that can transmit and receive data simultaneously, which implies a quite large amount of data recorded per second. Nonetheless, by implementing suitable procedures for a more efficient data handling, the problem has been successfully solved and now the prototype is not anymore "blind" in the field.

KEYWORDS: Ground penetrating radar; stepped-frequency reconfigurable systems; acquisition software; real-time data visualization.

1. INTRODUCTION

Pulsed and stepped-frequency systems are the two most widely used categories of ground penetrating radar (GPR) systems. They are based on ideas and principles dating back to the first half of the twentieth century [1]. According to [2], the first GPR technology patent was registered in 1910 and regarded a system working in the frequency domain, whereas the first pulsed system was patented in 1926, only. Nonetheless, the commercial development of GPR systems started after the Second World War (which gave a substantial input to the development of radar technology) and regarded pulsed systems, first; later, stepped-frequency systems were commercialised, too. In particular, while pulsed systems

 https://doi.org/10.26376/GPR2019003

were already commercialised in the sixties [1], the first experiments with commercial stepped-frequency systems date back to the seventies [3].

The debate on which system is best is still going on today [4, 5]. Stepped frequency systems are claimed to be more performant in terms of dynamic range and signal-to-noise ratio [4]. On the other hand, they present the problem that the receiver needs to have the same dynamic range as the transmitted signal, in order not to saturate when receiving the direct wave. Consequently, stepped-frequency technology is more complex, and this easily drives towards more expensive systems. However, the realization costs of pulsed and stepped frequency systems become similar for GPR systems with antenna arrays [6]. Probably, this is one of the main reasons why most commercial stepped-frequency systems are nowadays equipped with an array of antennas [7].

Although the majority of GPR systems currently are pulsed systems, we do not know in an absolute sense what is the best one between the pulsed and the stepped frequency technology. Probably, the answer to such a question also depends on the application. For example, according to [8] stepped-frequency systems are more promising for some high frequency applications such as demining. The possibility to reconfigure some hardware and software parameters during prospecting has been lately introduced for stepped-frequency systems [9]. In particular, a reconfigurable stepped-frequency system [10, 11] has been implemented within a research project called Aitech [12]. To the best of my knowledge, analogous reconfigurable pulsed systems do not yet exist.

Until recently, this reconfigurable stepped-frequency prototype had the problem that data were not visible in real time in the field. They were recorded in the field and could be viewed only during the post-processing. Real time data visualization is obviously an essential feature, which allows, e.g., checking the correct functioning of the system in the field and immediately identifying the presence of anomalies of interest in the area under test, so that a localized excavation can be carried out or further geophysical measurements can be made.

This paper summarizes the work carried out during my Master thesis in Computer Engineering, at the International Telematic University Uninettuno (Rome, Italy), in 2018, under the supervision of Dr Raffaele Persico (National Research Council of Italy, CNR, Lecce, Italy). My thesis was focused on developing new procedures to be integrated in the acquisition software of the above-mentioned reconfigurable stepped-frequency prototype, in order to allow real-time data visualization. In

 https://doi.org/10.26376/GPR2019003

Section 2, the prototypal reconfigurable GPR system and its original acquisition software are shortly described. In Section 3, the implementation of an improved version of the software is described, with a main focus on the procedures that have made it possible real-time data visualization during prospecting. Conclusions are drawn in Section 4.

2. THE STEPPED-FREQUENCY GPR PROTOTYPE

The prototypal GPR considered in this work is a stepped-frequency reconfigurable system, and in particular a stepped-frequency GPR where the three following parameters are programmable versus the frequency:

1. Integration time: Unlike any commercial stepped-frequency GPR, the reconfigurable prototype allows to set a different integration time for each frequency. This option allows to reject interferences (due, e.g., to external electromagnetic sources) without prolonging too much the acquisition time [13, 14].
2. Power attenuation: The reconfigurable prototype permits to program the power supplied to the transmitting antenna for each frequency. This option allows to equalize the shape of the synthetic pulse, if needed, and might be also used to avoid the saturation of the receiver at some frequencies.
3. Antennas: The reconfigurable prototype is equipped with a couple of shielded bow-tie antennas with two switches along each arm, as sketched in Figure 1. The switches are constituted of PIN diodes. By switching on and off the diodes, three equivalent couples of antennas can be achieved (“short”, “medium,” and “long”); hence, it is possible to change dynamically the geometry of the antennas. The system is capable to record three B-Scans simultaneously; to the best of my knowledge, commercial systems with dual antennas also exist, but they allow gathering two simultaneous B-Scans, at most. Moreover, in the case of the reconfigurable prototype at hand, all the equivalent antennas share the same gap, which customarily does not happen in commercial dual-band systems. The reconfigurable prototype can span the frequency range 50-1000 MHz three times (one for each equivalent antenna) at any measurement point, with a frequency step optionally equal to 2.5 or 5 MHz. The central frequencies of the three equivalent antennas are about 120, 250 and 520 MHz, for “long”, “medium” and “short” antennas, respectively.

https://doi.org/10.26376/GPR2019003

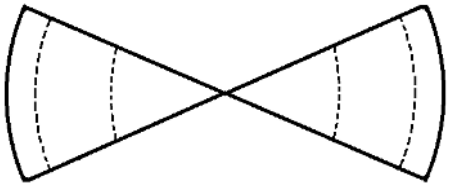

FIG. 1 – Scheme of the reconfigurable antenna. The antenna is "short" if all the switches (represented by dotted lines) are off, "medium" if the internal switches are on and the external switches are off, "long" if all the switches are on.

The original acquisition software of the reconfigurable prototype is called Eurydice. Eurydice configures the radar central unit according to settings defined by the user. In particular, according to the settings encoded by the user, the central unit will synchronize the aperture or closure of the antenna switches. Moreover, for each received signal the central unit will interrogate the odometer about the spatial position and save this information in the dedicated laptop computer along with the corresponding field data. Eurydice receives data packets at regular (uniform) time intervals; this, in general, does not correspond to regular spatial intervals because of the velocity variations of the human operator. Nonetheless, as the spatial positions are recorded with the data, the data can be suitably interpolated during post-processing to make the spatial step uniform. Data acquisition continues until Eurydice, following a command of the human operator, sends a stop request to the central unit.

3. UPGRADE OF THE ACQUISITION SOFTWARE

With the old version of Eurydice, the visualization of the data was possible only after collecting the data, i.e., in the post-processing stage. In particular, the visualization resulted from the execution of a homemade MATLAB script, which included the inverse fast Fourier transform for achieving time-domain data and an algorithm to perform spatial interpolation based on the information provided by the odometer.

As already mentioned in the Introduction, the lack of real-time data visualization was one of the main problems of the system, because the operator did not have an immediate picture of the investigated buried scenario and was not able to recognize in the field any possible incorrect functioning of the system.

 https://doi.org/10.26376/GPR2019003

The main reason why no implementation of real-time data visualization had been devised when the system was realized, was the large amount of data received by the application. It was necessary to implement a software capable to efficiently manage those data; this task was postponed to later.

Each 'burst' of received data is composed by a header and a data section. The header contains information about the kind of sweep and spatial position, along with some "free Bytes" reserved for possible future developments. The size of the header is 16 Bytes; the length of the data section, instead, depends on the configuration settings. In particular, each received harmonic signal is encoded with 8 Bytes times a programmable factor, which depends on the integration time. The default integration time is about 100 μs, whereas the larger available multiplier for the default integration time is equal to 10; hence, the longest possible integration time is 1 ms and maximum 80 Bytes can be used to encode the received signal at each frequency.

In each 'burst' of data all the harmonic signals received within the chosen frequency sweep are present. If the human operator chooses the largest available band, the frequency sweep starts at 50 MHz and stops at 1 GHz. If a frequency step of 2.5 MHz is chosen, this means that overall 381 different frequencies are considered, codified with 381*80 + 16 = 30496 Bytes, for each antenna. If data are recorded for all of the three available couples of equivalent antennas, Eurydice receives 91488 Bytes for every measurement position.

From the documentation provided by the constructor of the system we know that, with this configuration, the GPR sends bursts of data at the speed of 10.4 A-Scans per second (the A-Scan refers to the data recorded over a single measurement position). Consequently, in one second the application receives about 300 KB referred to all of the three antennas.

The application saves continuously the data in a file; however, the dedicated laptop computer has a limited computational power. In particular, the GPR system is equipped with an Asus EEPC computer, with a dual core processor Intel-Atom 2600, 1.6 GHz clock, 1.98 GB RAM and 300 GB Hard Disk; the operating system Windows XP is installed on the laptop computer.

The software upgrade presented in this paper is essentially based on the idea of discarding as soon as possible all data not strictly needed for a satisfactory real-time visualization (all data are still recorded on the

https://doi.org/10.26376/GPR2019003

hard disk of the laptop computer and can be subsequently visualized, analysed, and processed). This introduces a small degradation of the real-time image, but the loss of quality is acceptable.

The upgraded software asks the human operator to choose the antennas they want to select for real-time visualization. Indeed, during prospecting, the human operator can visualize on the screen the raw data related to one couple of antennas, only; however, it is possible to switch in any moment from a radargram to another one, among those being simultaneously recorded (i.e., it is possible to modify the initial choice and select a different couple of equivalent antennas for real-time visualization).

The real-time visualization requires also the management of an interpolation problem. In fact, as already mentioned, the acquisition rate of the system is constant versus time, not versus the covered distance, because the velocity of the human operator (or even of a mechanical vehicle) is not uniform with time. In particular, when the GPR system moves faster along the observation line, a higher computational power is necessary for the real-time visualization of the data. On the other hand, if the system moves very slowly, the visualization software receives redundant data, which can be partially discarded in order to allow real-time visualization.

The strategy for dealing with this issue is illustrated in Figure 2, where different possible cases are depicted. First, let us specify that the user selects a-priori a uniform spatial step for the visualization, so that we have a sequence of prefixed point, equally spaced, where we wish to

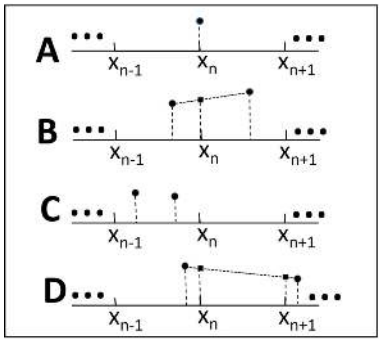

FIG. 2 – Sketch of the interpolation strategy implemented in the visualization software.

 https://doi.org/10.26376/GPR2019003

calculate the data to be visualized by performing an interpolation of the actual data. The true data are, essentially, electric-field amplitudes measured over points that we do not know a-priori, but that we record sequentially together with the field amplitudes.

In Figure 2, the point x_n is the n_{th} position where we wish to calculate the data to be visualized. In particular, the first datum is by definition gathered at the abscissa zero, but for the subsequent points, some discrepancy arises between the "desired" point next to x_n and the position where the data are actually measured.

The situation A in Figure 2 represents the fortunate but possible case when the n^{th} datum is gathered exactly in the n^{th} desired position. In such case, the datum is just processed for the visualization (let us remind that the system receives data in frequency domain, and so an inverse Fourier transform has to be performed in order to achieve representable real data in the synthetic time domain).

The situation B in Figure 2 represents the case when the last two gathered data are one at the left-hand side of x_n (after x_{n-1}) and the other at the right-hand side of x_n (before x_{n+1}). In this case, the visualization software applies a linear interpolation between the last two gathered data (accounting for their position, too) and the datum in x_n is easily retrieved (linear interpolation is applied after the inverse Fourier transform, hence in the time domain).

A third possibility is depicted on the third scheme of Figure 2 and is labelled as situation C. In this case, the last two gathered data both fall within the same interval, i.e., between the last "already-interpolated" point x_{n-1} and the next "to-be-interpolated" point x_n. In this case, the visualization code just discards the penultimate gathered value and keeps the last one. The C case can of course occur several times in a row, especially when the GPR system is moved slowly. Indeed, in this case, situation C will occur several times in a raw, and at a certain point situation B or, more rarely, situation A will happen.

The computationally most demanding situation is of course the fourth illustrated case (situation D). In fact, in this case, the penultimate point falls between x_{n-1} and x_n, whereas the last point is between x_{n+1} and x_{n+2}. In this case, the system applies a linear interpolation, too, but the missed points to be retrieved are two instead of just one, which requires a longer computational time. Obviously, if the "skip" is even larger, the visualization code could be also compelled to calculate three, four or "k" interpolated values, requiring a progressively longer time for the

 https://doi.org/10.26376/GPR2019003

interpolation. These larger interpolation calculations are needed when the system is moved too fast with respect to the spatial rate at which the data have to be interpolated. Of course, beyond a certain point, there is also a theoretical problem of underdamping of the field with consequent aliasing in the data, but at the typical velocity of a human operator this event customarily does not happen. The visualization code makes use of a buffer in order to compensate the difference of velocity between the acquisition rate and the elaboration rate, but if the system is moved too fast this buffer gets overcharged and the systems works not correctly, and eventually the acquisition process stops. On the other hand, if the velocity is very slow, the time needed for the prospecting increases and the files stored for the post processing are uselessly large.

Real-time data visualization is made further challenging for the system by the fact that the integration times of the harmonic components are not necessarily the same for each frequency. In fact, as explained in Section 2, the integration times are reconfigurable, and they can therefore have variegated lengths. Each integration tone, in particular, can be prolonged up to 10 times with respect to its default value [13]. However, in a real situation the multiplier parameter of the integration time will not be equal to 10 for all frequencies but only in the frequency intervals where noise or interferences are significant. For more details, the interested Reader is referred to [13]. Here, it is sufficient to say that an algorithm is able to program the ideal multiplication of the default integration time required for each harmonic tone, and then a threshold mechanism is applied to this calculation because for technological reasons the maximum value allowed for the extension of the prolongation factor is equal to 10, as already said.

The original Eurydice code was written in Vb.Net and made use of WinPcap libraries for internal communication [15]. For the software upgrade, the external ALGLIB [16] library was used too, in order to perform the inverse fast Fourier transform (IFFT) and deal more effectively with complex numbers.

The architecture of the upgraded code is schematixed in Figure 3. In particular, Figure 3 shows the data flow from the central unit of the GPR to the program, and in particular, the down pointing arrow indicates data packets rejected by the software for the real-time visualization. Then, the data useful for the visualization are collected in a ring buffer, where they are analysed; when necessary, they are sent to the graphical user interface (GUI).

https://doi.org/10.26376/GPR2019003

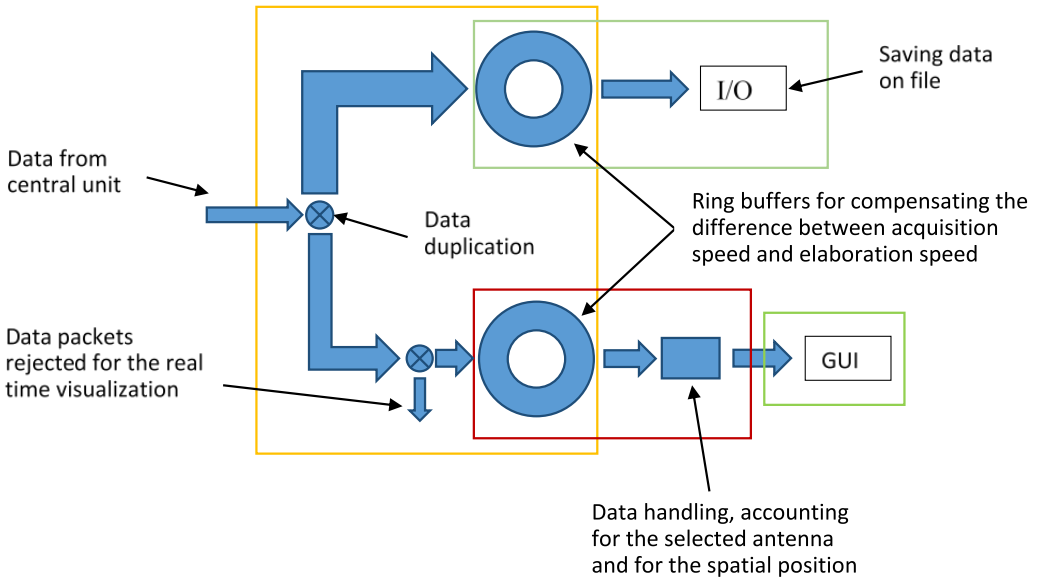

FIG. 3 – General scheme for the architecture of the upgraded Eurydice acquisition code. The upper branch was already implemented in the original software; the lower branch was added with the software upgrade.

Figure 4 is an alternative and expanded representation of the lower branch of Figure 3 (namely, the software upgrade). In particular, the various steps of the data-handling block are shown; these are:

- Recognition of useful data packets: The data packet is rejected if it comes from an equivalent antenna not selected for real-time visualization.
- Rejection of dead tone: This option was already present in the previous version of Eurydice. Not all of the measured harmonic data are saved, some of them are identified as “dead tones” and discarded. The dead tones are anomalous data not useful for the post processing and neither for the visualization.
- Management of integration times: Some averaging is done, to achieve the in-phase and quadrature components for each received harmonic tone.
- Spline interpolation: Possible edges in the received signal, due to commutations of the switches on the antennas, are smoothed.

 https://doi.org/10.26376/GPR2019003

- Calibration and upgrade of the calibration vector: The calibration data are not considered for the visualization, but they are used to normalize the data in the frequency domain.
- First positional logic block: This block correlates the elaboration of each data packet to its spatial position. In particular, it attributes the right position to the last datum that has to be transformed in the time domain.
- Inverse Fourier transform: This block retrieves data in the synthetic time domain. Within this step an optional zero padding can be performed in order to interpolate data, in the time domain, for the graphical visualization.
- Second logical position block: This block commands the inverse Fourier transform of the second datum needed to perform the interpolation (the first was acquired within the previous two steps). Then, this block gives start to the interpolation needed for the graphical representation.
- Interpolation and plotting: This final block performs spatial interpolation and video visualization according to the options set by the user, such as the contrast level and the colour palette.

In the portion of the graph included between dashed lines, in Figure 4, all data packets are handled in the same way. After this point, the operations made on the data are still the same, but the outputs are stored in different memory areas. From the user point of view, the main two differences between real-time visualization and post-processing visualization are the following:

- Only a limited spatial length that can be visualized in real time. In particular, in Figure 5 there is written “Max: 80 m”: in fact, after 80 m the software stops plotting. The distance of 80 m refers by default to a choice of 2 cm for the spatial step. Please note that this does not mean that the user has to stop the measurement after 80 m. If the B-Scan is recorded over a line longer than 80 m, data will not be visualized anymore after 80 m, but they will continue to be regularly saved and stored in the computer. The limitation of 80 m has been chosen because in this way the needed amount of memory is allocated only at the beginning and is the same for all the pictures (i.e., no dynamical allocation of memory is needed).

https://doi.org/10.26376/GPR2019003

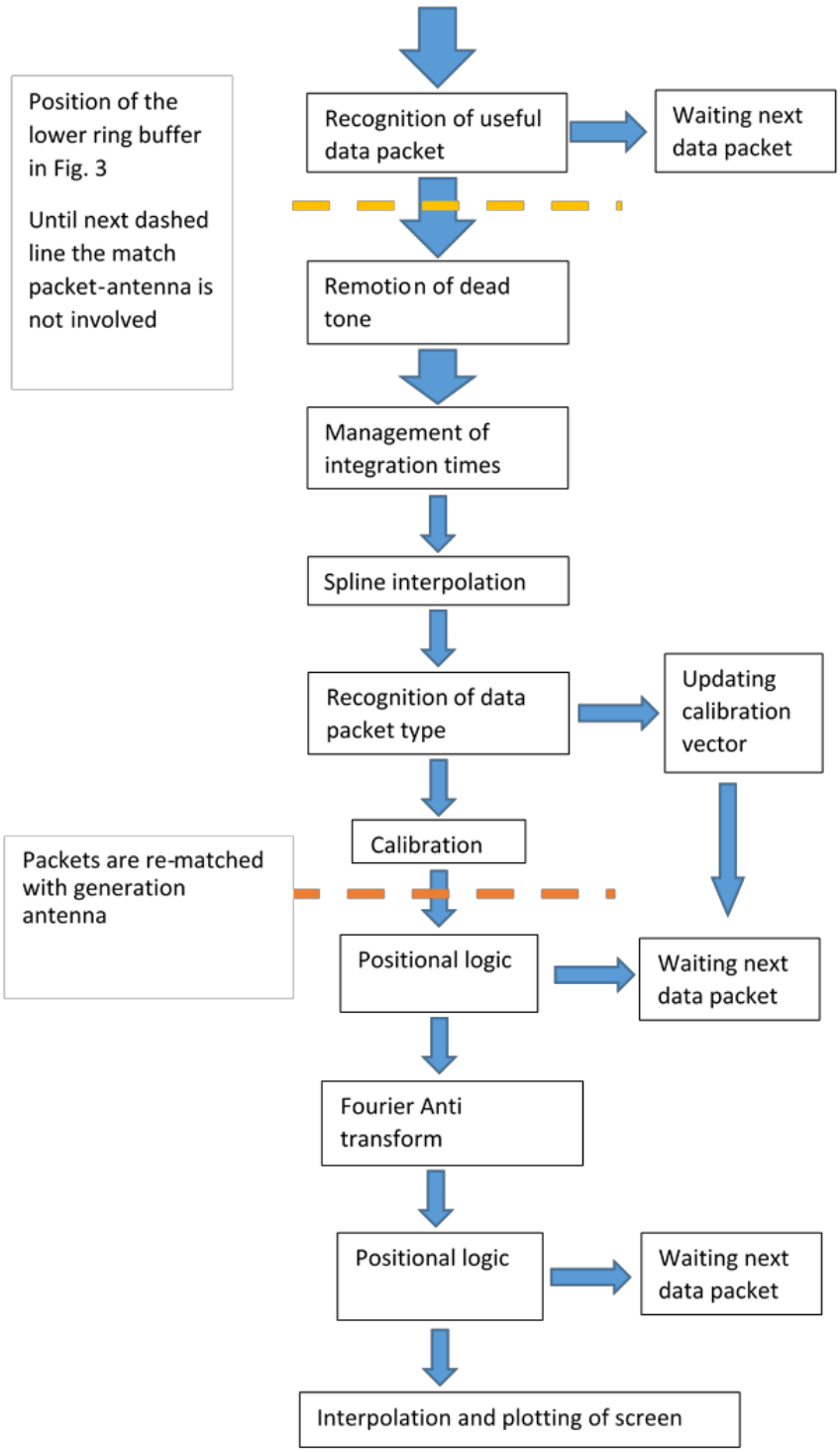

FIG. 4 – More detailled scheme of the lower branch of the diagram presented in Figure 3.

 https://doi.org/10.26376/GPR2019003

- The visualized time window has been chosen equal to 130 ns (without time interpolation), to guarantee a good readability of the graph. However, the bottom scale of the synthetic time domain data can be larger than this value, if the frequency step of the system is narrow enough to allow this. This means that the post-processed data can include also deeper time-depth levels.

The graphical user interface (GUI) of the new Eurydice software is shown in Figures 5 and 6. In particular, in Figure 5 two "panels" are presented; both these panels can be visualized by the user. The first panel is the interface for setting the parameters of the measurement; this panel was already present in the original version of the software. The second panel is available in the upgraded software, only; here, data are visualized in real time during the acquisition. In Figure 6, a screenshot of the second panel taken during a survey is shown.

For the interested Reader, the most interesting parts of the source code have been made available as 'Supplementary Materials' to this paper. ***Download Supplementary Materials here: http://gpradar.eu/onewebmedia/Supplementary_Materials_GPR-2-1-3_Brigatti.zip***

4 CONCLUSIONS

In this paper, a reconfigurable stepped-frequency ground penetrating radar (GPR) prototype was briefly presented, emphasizing some special features not common in commercial systems, such as the possibility to set a suitable integration time at different frequencies, and the availability of three couples of equivalent shielded antennas in the system, sharing the same gap. The main objective of the present paper, though, was to describe the development of new dedicated software procedures, allowing real-time data visualization in the prototype at hand; this is obviously a common feature in commercial systems, but was not yet available in the prototype. In particular, before the upgrade of the acquisition software presented in this paper, the visualization of the recorded (raw) data was possible only during the post processing stage, by using some Matlab scripts. The user can now set a spatial interpolation step and choose one or more antennas for real-time visualization during the prospecting, with huge practical benefits.

Future work might regard the development of a new reconfigurable stepped-frequency GPR (and relevant software) for higher frequency applications, such as through-the-wall prospecting, where the use of such a system is expected to be especially effective in contrasting external interferences.

https://doi.org/10.26376/GPR2019003

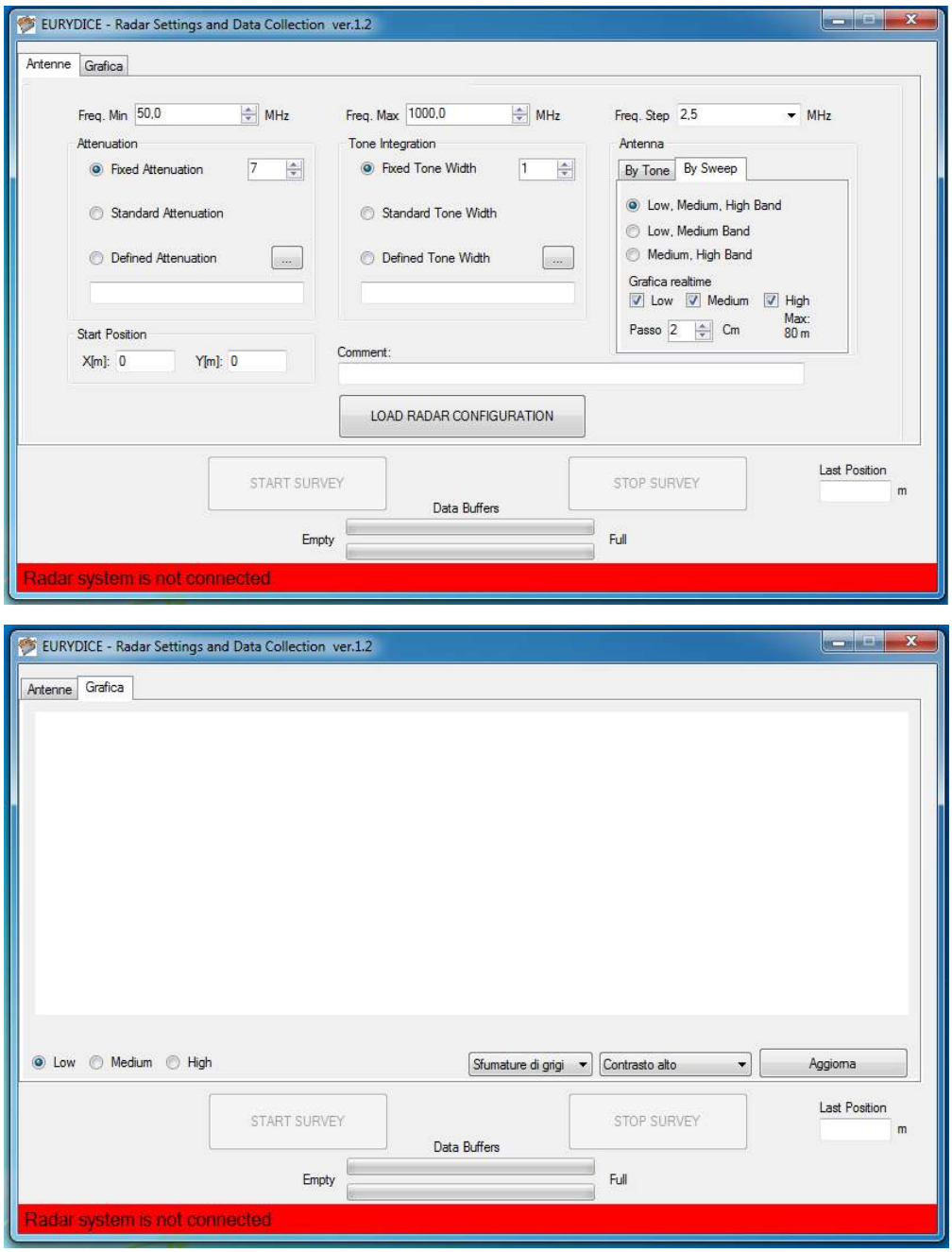

FIG. 5 – GUI of the upgraded software: data are visualized in real time on the panel shown in the lower part of the figure. Note that some terms are written in Italian: "Antenne" means "Antennas," "Grafica" means "Plot," "Sfumature di grigi" means "Greyscale," "Contrasto alto" means "High contrast," "Aggiorna" means "Refresh."

https://doi.org/10.26376/GPR2019003

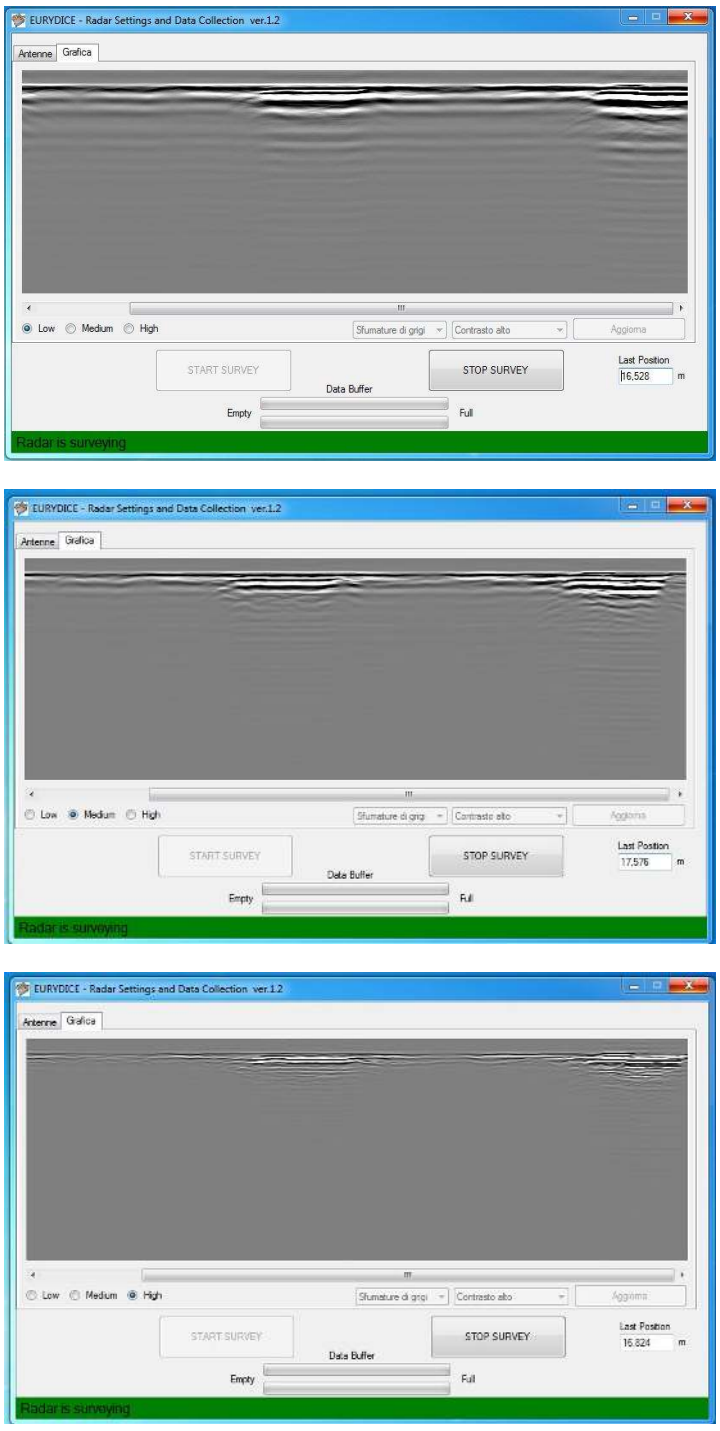

FIG. 6 – Screenshot of the software GUI during a real-field survey. Each picture shows a different B-Scan measured by a different equivalent couple of antennas.

 https://doi.org/10.26376/GPR2019003

Further possible research advancements might regard the design of novel reconfigurable antennas with an optimized shape, and also the development of new processing algorithms for a more advanced exploitation of data recorded by different antennas. Compared to commercial systems equipped with dual antennas, such exploitation is expected to be easier when dealing with the prototype presented in this paper, due to the fact that the equivalent antennas share the same gap.

Download Supplementary Materials here:
http://gpradar.eu/onewebmedia/Supplementary_Materials_GPR-2-1-3_Brigatti.zip

REFERENCES

[1] D. Daniels, Ground Penetrating Radar, 2nd Edition, IET, 2004.

[2] History of Ground Penetrating Radar (GPR): www.obonic.de/article/ground-peentrating-radar-history/ [Last visited on January 23rd, 2019]

[3] L. Robinson, W. B. Weirand, and L. Yung, "Location and recognition of discontinuities in dielectric media using synthetic RF pulses," Proceedings of the IEEE, vol. 62(1), pp. 36-44, January 1974, doi: 10.1109/PROC.1974.9383

[4] D. A. Noon, "Stepped-Frequency Radar Design and Signal Processing Enhances Ground Penetrating Radar Performance," Ph.D. Thesis, Department of Electrical & Computer Engineering, University of Queensland, Australia, 1996.

[5] G. Tronca, I. Tsalicoglou, S. Lehner, and G. Catanzariti, "Comparison of pulsed and stepped frequency continuous wave (SFCW) GPR systems: Applications on reinforced concrete and brick/rock masonries," Proceedings of the 17th International Conference on Ground Penetrating Radar, pp. 819-822, Rapperswil, Switzerland, 18-21 June 2018, ISBN 978-1-5386-5777.

[6] M. Sato and R. Persico, "Ground Penetrating Radar: technologies and data processing issues for applications," within the volume "Sensing the Past" edited by Nicola Masini and Francesco Soldovieri, Springer, Louisville, USA, 2017, ISBN 978-3-319-50518-3, chapter 9, pp. 175-202.

[7] E. Eide, N. Linford, R. Persico, and J. Sala, "Advanced SFCW GPR systems," in Innovative Instrumentation and Data Processing Methods in Near Surface Geophysics, edited by R. Persico, S. Piro and N. Linford, Elsevier, Amsterdam, The Netherlands, 2018, chapter 8, pp. 253-286, ISBN 978-0-12-812429-1.

[8] M. Sato, J. Fujiwara, and K. Takahashi, "ALIS evaluation tests in Croatia," Proceedings of the SPIE, vol. 7303, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XIV; 73031B (2009), doi: 10.1117/12.818497.

 https://doi.org/10.26376/GPR2019003

Event: SPIE Defense, Security, and Sensing, 2009, Orlando, Florida, United States.

[9] R. Persico and G. Prisco, "A Reconfigurative Approach for SF-GPR Prospecting," IEEE Transactions on Antennas and Propagation, vol. 56(8), pp. 2673-2680, August 2008, doi: 10.1109/TAP.2008.927516.

[10] R. Persico, M. Ciminale, and L. Matera, "A new reconfigurable stepped frequency GPR system, possibilities and issues; applications to two different Cultural Heritage Resources," Near Surface Geophysics, vol. 12(6), pp. 793-801, December 2014, doi: 10.3997/1873-0604.2014035.

[11] L. Matera, R. Persico, N. Bianco, G. Lepozzi, and G. Leopizzi, "Joined interpretation of Buried Anomalies from Ground Penetrating Radar data and endoscopic tests," Archaeological Prospection, vol. 23(4), pp. 301–309, September 2016, doi: 10.1002/arp.1545.

[12] Website of the A.I.Te.C.H. project: www.aitechnet.com [Last visited on January 23rd, 2019].

[13] R. Persico, D. Dei, F. Parrini, and L. Matera, "Mitigation of narrow band interferences by means of a reconfigurable stepped frequency GPR system," Radio Science, vol. 51(8), pp. 1322-1331, August 2016, doi: 10.1002/ 2016RS005986.

[14] R. Persico and G. Leucci, "Interference Mitigation achieved with a reconfigurable stepped frequency GPR System," Remote Sensing, vol. 8, pp. 1-11, Article ID 926, November 2016, doi: 10.3390/rs8110926.

[15] Website of the WinPcap library: www.winpcap.org [Last visited on January 23rd, 2019].

[16] Website of the ALGLIB project: www.alglib.net [Last visited on January 23rd, 2019].

The scientific paper that you have downloaded is included in Issue 1, Volume 2 (March 2019) of the journal *Ground Penetrating Radar* (ISSN 2533-3100; journal homepage: www.gpradar.eu/journal).
All *Ground Penetrating Radar* papers are processed and published in true open access, free to both Authors and Readers, thanks to the generous support of TU1208 GPR Association and to the voluntary efforts of the journal Editorial Board. The publication of Issue 1, Volume 2 is also supported by IDS Georadar s.r.l. (idsgeoradar.com).
The present information sheet is obviously not part of the scientific paper.