SOFTWARE CONSIDERATIONS FOR ULTRASONIC DATA COLLECTION AND EVALUATION SYSTEMS

L. Svilainis, L. Sinkunas

Signal processing department, Kaunas University of Technology
Studentu str. 50, LT-51368 Kaunas, Lithuania
T, +370 37 300532; F, +370 37 753998;E, linas.svilainis@ktu.lt / linas.sinkunas@delfi.lt.

Abstract

The ultrasonic data acquisition and analysis system software concept is analyzed. Modular software configuration is suggested. It should be capable to adapt to new hardware configurations. The core-based structure is suggested. External units planned are: acquisition, imaging, processing, analysis and tools. Acquisition unit is for hardware control, data exchange and communication. Structure is controlled by separate software and core is only using the defined DLL module. Imaging unit is responsible for visualization of data as line plot A-scan, color coded, mesh, surface or contour plot of B- or C-scan, volume rendering. Graphic user interface (GUI) is planned to be twofold. One part of GUI resides in core such as menu, toolbars, views, controls and status bar management. Remaining parts of GUI are coming together with external units. Only essential for the current state controls are displayed so user is not disrupted by variety of buttons and whiskers.

1. Introduction

Nowadays many applications are using ultrasound. It can be used in technical diagnostics, navigation, security, or in medical diagnostics and treatment.

Large variety of measurement devices are created for this purpose. However, market is missing not expensive, portable, yet sufficiently functional and flexible ultrasonic measurement system dedicated for scientific research. Such system should be configurable and with possibility to extend and adapt to different scientific or commercial applications.

Hardware of such system is already under development [1]. This hardware will need the modern software. The aim of this publication is to analyze the modern design techniques and concepts of software dedicated for data acquisition and processing. The software considerations for ultrasonic data collection and evaluation system should be rectified.

2. Software architecture

Currently existing commercial frameworks with image processing and visualization capabilities [2] include Amira, Analyze, AVS[3], IDL[4], IRIS Explorer, Matlab [5]. Commercial versions of VolView [6], and MeVisLab [7] are also available. But commercially available frameworks are expensive and not suitable for budget application that we target. In this case we consider that would be the best to use popular freeware open source libraries or frameworks [2,9,10]. It can be MITK, VolView, MeVisLab or SCIRun [2]. MITK is oriented for end-user applications with fixed algorithm configurations. VolView is the best for non-repetitive end-user applications based on existing functionality. MeVisLab and SCIRun are the best for creating applications prototypes using visual data-flow programming.

However all of them can not satisfy our needs because they have already defined user interface and limited number of algorithms and functionality. Therefore decision was taken to write our framework on the base of ITK and VTK libraries using the C++ language as glue and QT library for GUI (graphical user interface). Suggested tools hierarchy is in Figure 1.

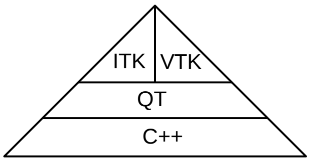

Figure 1. Tools hierarchy

The justification for such decision is:

– libraries are open source therefore are wanted for budget applications;

– they are multiplatform, permanently under development and already popular in academic community so the availability of solutions similar to ours is high.

The software architecture structure is presented in Figure 2.

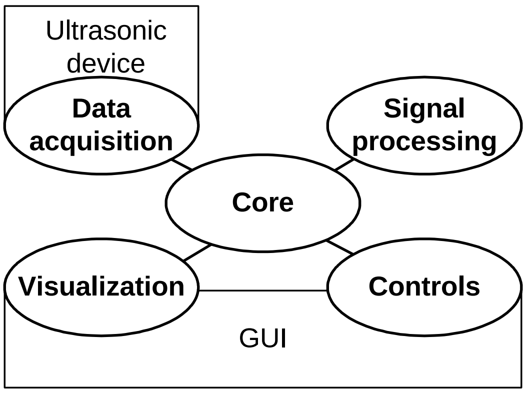

Figure 2. System architecture

The top agent in PAC hierarchy is Core that is mostly responsible for organizing connections and data exchange between different agents. The subagents can be distributed into several groups: Data acquisition (DA), Signal processing (SP), Visualization (Vi) and Controls (Ctrl).

2.1. Core agent

Core is responsible of control of underlying agents and their interaction. This agent also builds GUI screens and widgets into composition. The whole hierarchy of the application is composed according to Presentation – Abstraction – Controller design pattern where each agent of the hierarchy is responsible for certain functionality of the system and is implemented according to Model –View - Controller pattern. All the units are derived from base unit class single for every agent.

2.2. Data acquisition agent

This agent is responsible for data acquisition from ultrasonic device. The structure ultrasonic device is shown in Figure 3 [11].

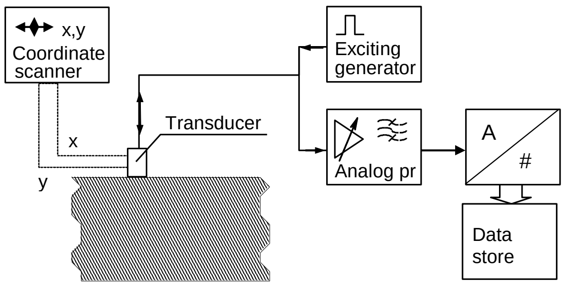

Figure 3. Hardware structure

The connection with external hardware is accomplished via USB interface, but the possibility to use other data collection and control interfaces as well as other USB device types is foreseen. For this purpose software acquisition agent must be derived from base data acquisition agent class. Hardware structure is described and DA agent structure is controlled by separate software and Core is only using the defined DLL module.

During acquisition process DA agent work as separate thread in parallel with other threads. All the data obtained for the hardware is stored in agent's data buffer. We suggest using client – server computing architecture which separates a client from a server. Suggested structure of workflow is presented in Fig. 4.

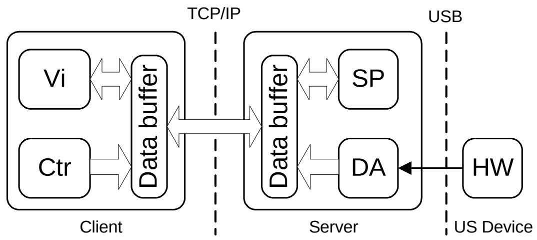

Figure 4. Workflow structure

The data acquisition and signal processing agents are located on server side. The server always takes data from data acquisition device, makes signal processing and sends to client by request.

The visualization and controls agents can be located at client side. The Core agent resides both on client side and on application side since it is responsible for task synchronization and GUI which interacts with user.

Such workflow structure allows to be realized in simple way, when server and client are in one single process. Still they work as separate threads but use common memory. Workflow can be made more complex, when server and client threads are working as separate processes on different computers. Then number of servers can be increased, allowing for multiple data source management (Figure 5).

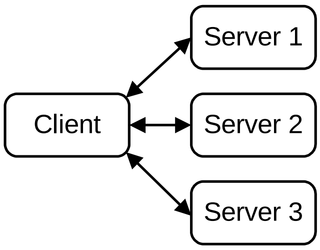

Figure 5. Multisource manager

The communication can be done by using TCP/IP connection increasing the remote capabilities of the system.

2.3. Signal processing agent

Signal processing agent is responsible for signal processing. Notable that thanks to template based data interface signals supplied for the same processing routine can be 1D, 2D 3D–dimensional. It can be used as standalone processing unit with individual data input and output nodes and parameters node. Or, single SP units can be combined to create more complicated processing branches (Figure 6).

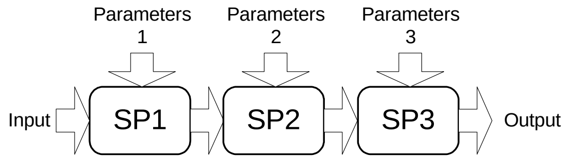

Figure 6. Chaining of signal processing units by forced execution

Every unit has one or more inputs and outputs and one or several parameter inputs. Several units can be combined into serial or split branch. Every unit has its own GUI for parameters and can be stored in separate DLL file. Same is applicable for all agents.

The execution of split branches is complicated. By default it is processed forward. But execution on request principle can be applied. Such approach is extremely useful if algorithm has a lot of branches and is time time-consuming. Then unit should be made of two parts: processing and run manager (Figure 7).

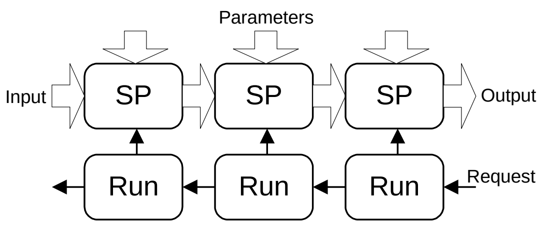

Figure 7. Signal processing unit chain with requested execution

Chained units can pass the processing request up to the desired branch, so whole tree update is not necessary. Here we present the example from the ITK library:

filter2->SetInput(filter1->GetOutput())
filter3-> SetInput(filter2->GetOutput())
filter3-> Update()

Predicted units for SP agent:
– FFT
– Filters;
– Arithmetic operations;
– Morphological operations;
– Cross-correlation processing;
– Radon transformation;
– Segmentation (thresholding);
– Averaging (moving average);
– Wavelet processing;
– Color coding;
– Detector (peak, min, max);
– Geometry measurement.

2.4. Visualization agent

Visualization agent is responsible for data imaging. Since data can be of different dimensionality, it is necessary to have the corresponding imaging possibilities. The most popular imaging modes are presented in Figure 8.

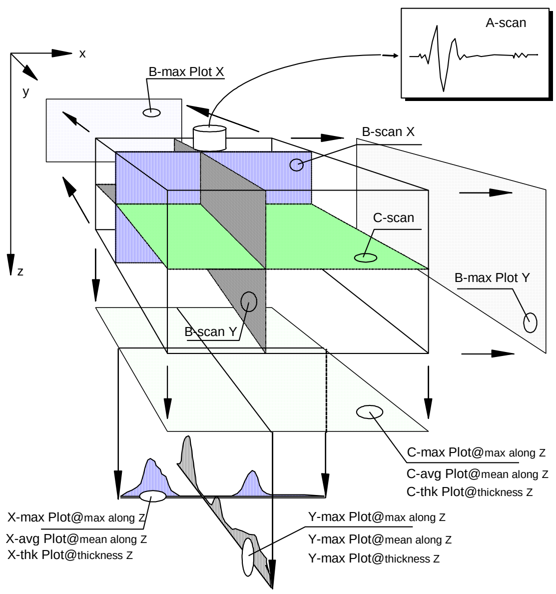

Figure 8. Possible imaging modes

In general visualization can be grouped in 1D, 2D and 3D modes. The 1D imaging example is an A-scan (Figure 9).

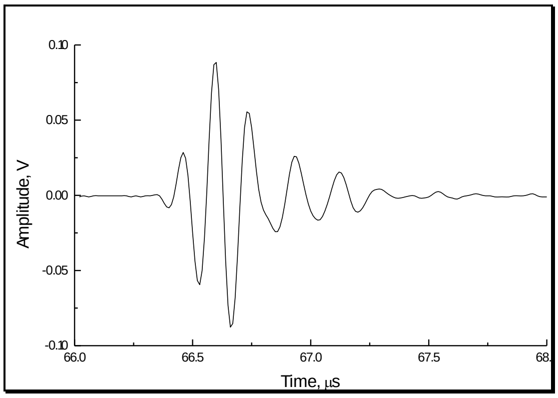

Figure 9. Line plot of received signal (A-scan)

The two most popular 2D imaging modes are B-scan (Figure 10) and C-scan (Figure 11).

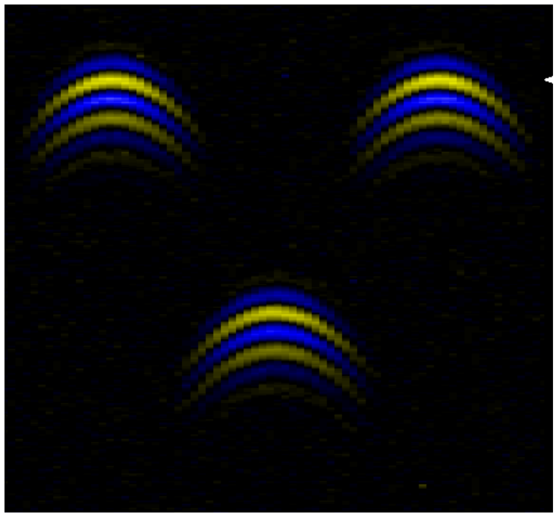

Figure 10. B-scan of 3 reflectors

Another subdivision of 2D images can be grouped into:
– color coded;
– grey scale;
– mesh;
– surface lighting;
– contour plot;
– or can be a combination of mentionned above.

In general 2D images can be obtained also from other sources such as line scan or matrix optical cameras, Xray devices. Software should be capable of multiple data source data fusion both coordinate and imaging sense.

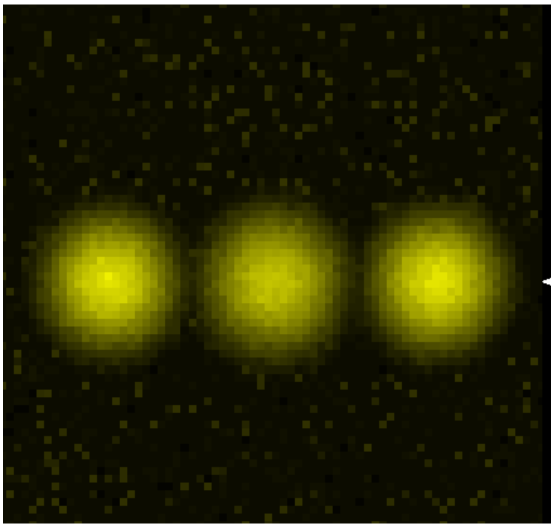

Figure 11. C-scan of 3 reflectors

The most attractive imaging mode is 3D (Figure 12 [12]).

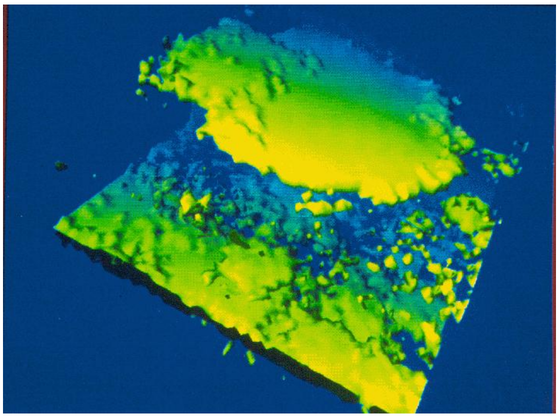

Figure 12. 3D-scan

3D scan is volume visualization and needs more processing time. Now with new computer performance, 3D image processing and visualization becomes more and more popular.

There are two types of 3D visualization:

– segmentation;

– volume rendering.

Segmentation is the application of synthetic geometric primitives which are obtained from thresholded volume objects.

Volume rendering is more realistic, but more difficult for human eye perception. Very often a combination of segmentation and volume rendering is used.

In our software 1D, 2D and 3D visualization will be realized as screen agent derived from base screen agent class.

2.5. Controls agent

Controls agent is responsible for setting and passing the parameters to units. These tools are planned into new software:

– image navigation;

– distance, angle measurement;

– algorithm execution;

– absolute coordinates measurement;

– subvolume selection.

3. Conclusions

The described software architecture allows for budget and fast software development for ultrasonic data acquisition system. Flexible software configurability is embedded. Such architecture allows connecting another type of data acquisition equipment such as optical, infrared cameras, X-raying devices etc.

Simple software configuration can be easily grown for more complex tasks.

References

[1]. L.Svilainis, G.Motiejunas, "Data acquisition system for air-coupled navigation study: the concept", Ultragarsas, No.2(62), 2007, pp.21-25.

[2]. I.Bitter, et.al. "Comparison of Four Freely Available Frameworks for Image Processing and Visualization That Use ITK", IEEE Transactions on visualization and computer graphics, Vol. 13, No. 3, pp.483-493.

[3]. Advanced Visual Systems, AVS, http://www.avs.com, 2006.

[4]. The Data Visualization & Analysis Platform, IDL, http://www.rsinc.com/idl, 2006.

[5]. MatLab, http://www.mathworks.com, 2006.

[6]. VolView: A Volume Visualization System, Kitware Inc., http://www.VolView.com, 2006.

[7]. MeVisLab: A Development Environment for Medical Image Processing and Visualization, MeVis, http://www.MeVisLab.de, 2006.

[8]. J.Blanchette, M.Summerfield, "C++ GUI Programming with Qt 4", Prentice Hall, 2006.

[9]. W.Schroder et.al., "The visualization toolkit: an object oriented approach to 3D graphics", 4-th edition, Kitware, 2006, p.512.

[10]. L.Ibanez et.al., "The ITK Software Guide", Insight, 2005. p.836.

[11]. L.Svilainis, V.Puodziunas, "Ultrasonic NDE system: the hardware concept.", Ultragarsas, No.29, 1998, pp.34-40.

[12]. L.Svilainis, N.Saffari, H.Mahlouji, "Signal processing for ultrasonic imaging of composites", conf. proc. Inspection of structural composites, Bentham Press, London, 1994, pp.138-153.