# Spatial and temporal multiscale interactive visualization: two prototypes

Debora Testi[1], Daniele Giunchi[1], Julien Finet[2], Stephen Aylward[2]

[1]*SCS, Italy;* [2]*Kitware Inc, USA*

Correspondence: d.testi@scsitaly.com, Via Parini 1, 40033 Casalecchio di Reno (BO), Italy

## 1. Introduction

Multiscale interactive visualization with biomedical data has started to receive attention in the Virtual Physiological Human [1] community as the first results and data from integrative research are becoming available. To tackle this multifaceted and demanding challenge, a consortium of international stakeholders in biomedical data visualization has been set-up to implement an open-source software library for the multiscale data visualization [2].

The activity started with a definition of the challenges to be addressed, the collection of exemplary problems and the identification of best practices, which led to a complete formalization of the multiscale visualization problem [3]. Based on the achieved understanding of the issues to be addressed, the multiscale interaction paradigm has been defined and the implementation of the software library, MSVTK, has started. To demonstrate the effectiveness of the proposed approach demonstrators are being developed based on exemplary problems: the two prototypes in most advanced stage of development are here presented.

## 2. The interaction approach

The interaction paradigm, which has been proven to be very effective in other domains, such as geography (for example in Google Earth), and which was already experimented with the EU-funded project LHDL [4] has also been adopted in the MSVTK library: it is the click-and-zoom interaction approach.

With this specific paradigm, visual placeholders allow the identification of the position of lower scale data with respect to the whole scene, providing an easy interface for data navigation. However, to be really effective, this should be associated to a way for the user to understand the relative positions of the various data (also considering that some might be not visible at the current scale): a data tree (to be used also as alternative data selection tool), or a map on the corner showing which part of the overall scene the user is actually looking at can be considered for this purpose. Moreover, visual cues might be used not only to indicate the physical position of the data but also to convey meaningful information about the data itself (i.e. the data type). A clever use of the icons shape and colour can be used to optimize the information provided to the users: for example, additional information on the represented data might be provided including in visual cues hyperlinks to extra documentation, etc. An extension to basic VTK functionalities to deal better with time-varying data and/or information is also being added.

## 3. MSVTK library

The above described interaction paradigm is being implemented as a software library containing elements, as widgets, that might be used at the end of the project also in other software development projects to add support to the multiscale visualization.

The MSVTK library is being implemented in C++, as an extension of VTK (Visualization ToolKit) [5]. Because of the innovative nature of MSVTK development, the MSVTK team uses the latest versions of VTK, CTK [6], CMake [7], and Qt [8]. Moreover the core of the library is being designed to be as much as possible platform independent, thus compiling under Windows, Linux, and Mac OSX operative systems. In particular, to implement the visual cues concept MSVTK relies on vtkButtons elements, which are here extended to the purpose of multiscale visualization of biomedical data. These “buttons” are represented by a geometry that can be textured and are divided into two regions: the texture and the shoulder regions. The points in both areas are assigned texture coordinates. The texture region has texture coordinates consistent with the image to be placed on it. All points in the shoulder regions are assigned a texture coordinate specified by the user. In this way the shoulder region can be coloured by the texture.

As already mentioned, the MSVTK components are being used in the development of prototypes on the exemplary problems collected, so to allow checking the efficacy of the proposed approach. Moreover, in order to verify the generality of the implementation, in the development of the different demonstration applications, the MSVTK library tools are being integrated into other frameworks such as the Multimod Application Framework (MAF) [9], and VTK itself.

## 4. Spatial multiscale demonstrator

The aim of the first prototype is to verify the use of vtkButtons to manage the chosen click-and-zoom interaction paradigm when dealing with spatial multiscale data. The aim is to efficiently and effectively visualize multiple datasets inside a MAF-based application, which uses vtkButtons as an interactive tool placed in a mafView. The prototype began with re-implementing the LHDL use case but progressing to overcome its original limitations. In particular, the user is allowed to navigate 3D image datasets at different spatial scales (CT scan at organ level, microCT scan, and nanoscan) making use of the vtkButtons both to move from one scale to another and also to zoom back to the whole system scale into a slice view (Figure 1). The property of the vtkButtons to always remain in the scene, even when the data is too big to be visualised, has been found effective also to zoom back to a higher scale, while in the LHDL prototype GUI elements had to be created to the purpose complicating the user interface.

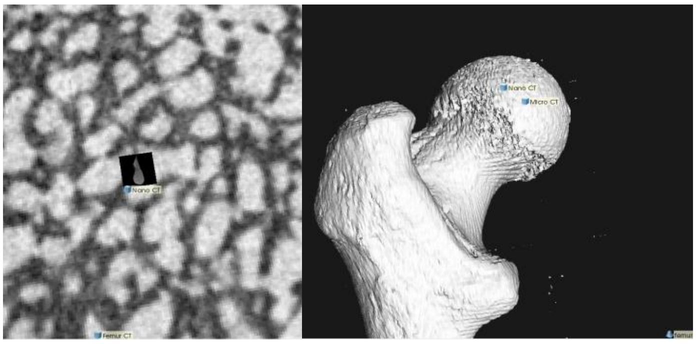

Figure 1 – Prototype interface of click and zoom interaction: the micro and nano datasets in a slice view with vtkButtons of the large scale femur available to zoom out (slice view with volume data on the left), and with mixed data types (surfaces and volumes) and visualization modalities (3D rendering and slice) (on the right)

To allow easy testing of different configurations, some features of the vtkButtons behaviour have been exposed in the GUI (Figure 2), such as visibility, label, position with respect to the bounding box, and interaction style (zoom or fly to), and more properties will be added in the future like the possibility to set the vtkButtons. This will allow optimization of the interaction behaviour but leaving the possibility to change it to satisfy specific user scenarios needs.

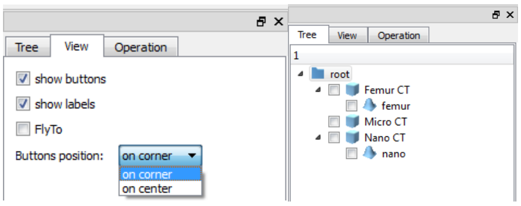


Figure 2 – vtkButtons properties exposed on the left and data tree to provide an overview of the scene on the right

This prototype already shows a number of advantages with respect to the previous LHDL prototype. For example, the navigation can be easily extended to other data types besides volumes, with a mixed visualization between surfaces and volumes already tested (Figure 1, right). Moreover, the advantage of having the visual cue visible, even if the lower scale data is hidden by a bigger object, allows to manage occlusion. vtkButtons seem also very promising for smooth data navigation when not all scales are available. Further work is being pursued to include other data types and explore the visualization modalities most suited to other exemplary problems to check the generality of the approach.

## 5. Temporal multiscale demonstrator

The second prototype has been implemented in order to manage and interact with multiscale temporal data and developed using Qt, CTK, and VTK. The demonstrator is in this case addressing a cardiological domain scenario with the heterogeneous data type, sparse data, and time scale issues (Figure 3). Sparse points on a 3D heart surface are displayed in a VTK render window. To each point, ECG data are registered, which are varying values over a certain time frame. When selecting one of the points, the ECG(s) is visualised in a separate VTK chart where time is represented differently depending on the data. To manage the navigation into different frames for each data a traditional slider cannot be used in the GUI. A new time player widget has been developed (Figure 4): it is a Qt widget that controls the time information of a VTK filter. By connecting the widget with a VTK mapper, it extracts the time range information from the VTK pipeline. On user interaction (play, seek, backward/forward step...), it propagates the current time index of the filter upstream in the pipeline. The application listens to the Qt time player widget using a Qt signal/slot connection and refreshes the VTK render views on time change.

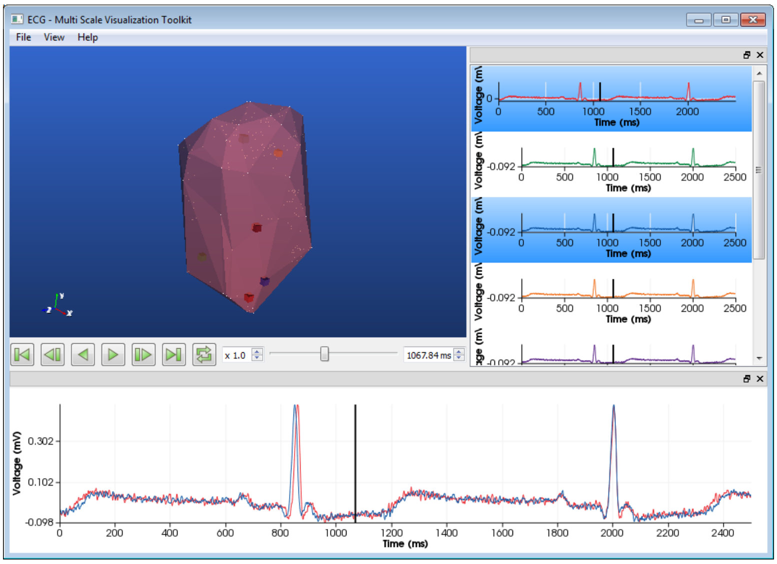


Figure 3 – Second prototype interface for the multiscale visualization of cardiological data

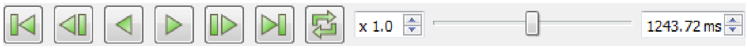


Figure 4 –MSVTK Qt time player widget

## 6. Conclusions

MSVTK is trying to cover the important aspect of the interactive visualization, which is still missing in the available software frameworks dealing with the visualization of biomedical data: the user friendly and efficient interaction with data at different spatial and temporal scales. The interaction approach has been defined based on a number of exemplary problems, which have been collected worldwide, analysed, and made public. The MSVTK library is being tested with the implementation of prototypes, which aim at demonstrating the efficacy of the proposed solution in different biomedical contexts. Even if various aspects of the multiscale interaction are presented for seek of clarity in separate applications, nothing will prevent in the future to use a combination of them in order to satisfy the specific biomedical need.

Even if the development is still on progress, the chosen approach and implementation has been so far very effective when dealing with the first set of use cases. Different use scenarios and prototypes will be analysed in the next year to identify and address other specific issue. By end of 2012, MSVTK library will be released in open source and made available to the biomedical community at large.

## Acknowledgements

This work was partially supported by the Virtual Physiological Human Programme of the European Commission under the MSV project (FP7 #248032).

## References

1. Virtual Physiological Human, http://en.wikipedia.org/wiki/Virtual_Physiological_Human

2. Testi D, Clapworthy G, Planes X, Christie R, and Aylward S. VPH challenges: a solution to interactive visualisation of biomedical data, VPH2012 conference, accepted, 2012.

3. Multiscale visualisation White Paper, https://www.biomedtown.org/MSV/reception/repository/MSV_D22_final.pdf

4. McFarlane NJB, Clapworthy GJ, Agrawal A, Viceconti M, Taddei F, Schileo E, and Baruffaldi F. 3D Multiscale visualisation for medical datasets, Proc. 5th Int. Conf. on Biomedical Visualisation, MediVis08, 47-52, London, UK, 8-11 July 2008.

5. Visualisation Toolkit, http://www.vtk.org

6. CommonTk, http://www.commontk.org

7. CMake, http://www.cmake.org/

8. Qt, http://qt.nokia.com

9. Multimod Application Framework, http://www.openmaf.org